International Journal of Networks and Communications 2012, 2(6): 155-169
DOI: 10.5923/j.ijnc.20120206.04

Middleware Services for Network Disruption Tolerant Mobile Applications

Sangwhan Cha*, Weichang Du

Faculty of Computer Science, University of New Brunswick, Fredericton, E3B 5A3, Canada

Abstract The demand for mobile application services has increased recently. However mobile wireless access technologies so far are not mature enough to address concerns with the reliability of mobile services. Therefore, mobile services are prone to failures caused by disruption of active wireless access network connections due to device movement. It is important to have proper mechanisms for managing mobile application services in case of loss of network connections. In this paper, we propose a middleware services that transparently performs required functionality to users in order to provide efficient mobile services in case of network disruption. Such middleware service provides an effective disconnection tolerant mobile application services.

Keywords Middleware, Mobile Application, Mobile Service, Network Disruption Tolerant, System Architecture

1. Introduction

Mobile computing is a computing environment allowing a mobile device to connect to a network wirelessly while moving freely. It is expected to become the future of computing technology. Stationary and mobile computing are distinct because the limited resources on mobile devices and the capacity of wireless network connectivity gives mobile devices a different focus. Because of these distinct features, the role of networks should be considered carefully when designing and implementing mobile application services. In addition, a reliable wireless connection has to be guaranteed to provide mobile application services properly.

The demand for mobile application services has increased recently. However mobile wireless access technologies so far are not mature enough to address concerns with the reliability of mobile services. Therefore, mobile services are prone to failures caused by disruption of active wireless access network connections due to device movement. It is important to have proper mechanisms for managing mobile application services in case of loss of network connections.

In this paper, we investigate a novel architecture for network disruption tolerant mobile applications, providing effective and efficient management for mobile services, network connections, resources and context handling.

The rest of the paper is structured as follows. Section 2 introduces current related work. Section 3 discusses the mobile service problem. Section 4 introduced our proposed middleware solution. Section 5 explains detailed context aware decisions. Section 6 and section 7 describe our experiments and evaluation respectively.

Finally, Section 8 gives some concluding remarks.

2. Related Work

There have been some related works for disruption tolerant concerns with regard to application layer approach and Disruption Tolerant Network (DTN). Some approaches on application layer have been dealing with content modification on servers and others have been dealing with wireless applications within a mobile ad-hoc network (MANET). Most approaches[1-5] on DTN have been dealing with end to end routing technology.

Liu et al.[6] propose disruption-tolerant content-aware video streaming using the notion of substitutable content summary frames and a method to select these frames. In this approach, summary frames are sent in advance to be displayed during disruption period. Summary frames, which are represented using shaded rectangles, are selected with the algorithm based on content difference of all adjacent frames in the list. After selecting summary frames, all frames are rearranged so that summary frames could be sent ahead of complementary frames and be displayed during network disruption period.

Data loss, which can be effective for video quality, during disruption period and data handling, which is related to the provision of data in server side, after network recovers were not considered. Video sequence with additional summary frame has to be rearranged.

Moghadam et al.[7] propose a modular platform to develop mobile disruption-tolerant applications based on

* Corresponding author:
scha@unb.ca (Sangwhan Cha)
Published online at http://journal.sapub.org/ijnc

previously proposed platform: 7DS which was developed to address network disruption problem in mobile networks by providing store-carry-forward communication. 7 DS stands for "Seven Degrees of Separation"[8].Applications, which need file sharing and synchronization, are considered as mobile applications. Sharing document modification with others, potable classroom system and Bulletin Board System (BBS) between teacher and students are examples as mobile applications in this paper. 7DS architecture mainly consists of discovery module, data sharing, and delta encoding. Discovery module enables mobile users to create an IP network automatically without any server such as Dynamic Host Configuration Protocol (DHCP) or Domain Name System (DNS) and helps 7DS users in the physical proximity to announce and locate the presence of services so that all services over node are registered and advertised[7]. Date sharing module enables mobile peers to share and synchronize data with file synchronization module and bulletin board system module which is responsible to exchange data in push-based mode. Delta encoding module is responsible to make a more efficient use of network bandwidth by transferring a minimal description of changes instead of the entire new instance of the resource. Although 7DS is a platform that serves as an environment to develop mobile disruption-tolerant applications, it only concerns data as store-carry-forward manner for its own defined disruption-tolerant applications, which are not true mobile applications.

Kopena et al.[9] propose service-based computing for agents, which act as proxies for the sender, on disruption and delay prone networks. The domain challenges of proposed approach are latency, bandwidth, link errors, asymmetric links, low connectivity, and others. Mobile agents and services are main used technologies. Satellite relays scenario has been made with some arguments such as optimal networking, effective agents, reason, etc.. Since this approach is similar to store and carry method using an agent technology without any data loss, it does not provide any solution for real time mobile applications.

3. Mobile Service Problem

As stated in[10], we may categorize many different kinds of mobile applications into mobile multimedia applications, mobile interactive applications and mobile web applications based on data communication patterns.

Mobile applications that are running on mobile devices need mobile services provided by servers in order to be played as mobile multimedia applications, mobile interactive applications, and mobile web applications. In this paper, we use the term mobile application to mean software running on mobile devices, and mobile service to mean a necessary support provided by servers for mobile multimedia, interactive and web applications.

There are two kinds of wireless access network disconnections: sudden disconnection and gradual disconnection. Sudden disconnection is caused by signal blocking while gradual disconnection is caused by device movement to other networks. Figure 1 shows a gradual disconnection scenario between base station 1 and base station 2 when a mobile device moves from point (a) to point (d). Mobile application services may not be provided properly during network disconnection from point (b) to point (c) with the undesirable termination of mobile application services. In this paper, we consider that network disruption is caused by the situation in which no access network is available, due to mobile device movement or signal blocking.

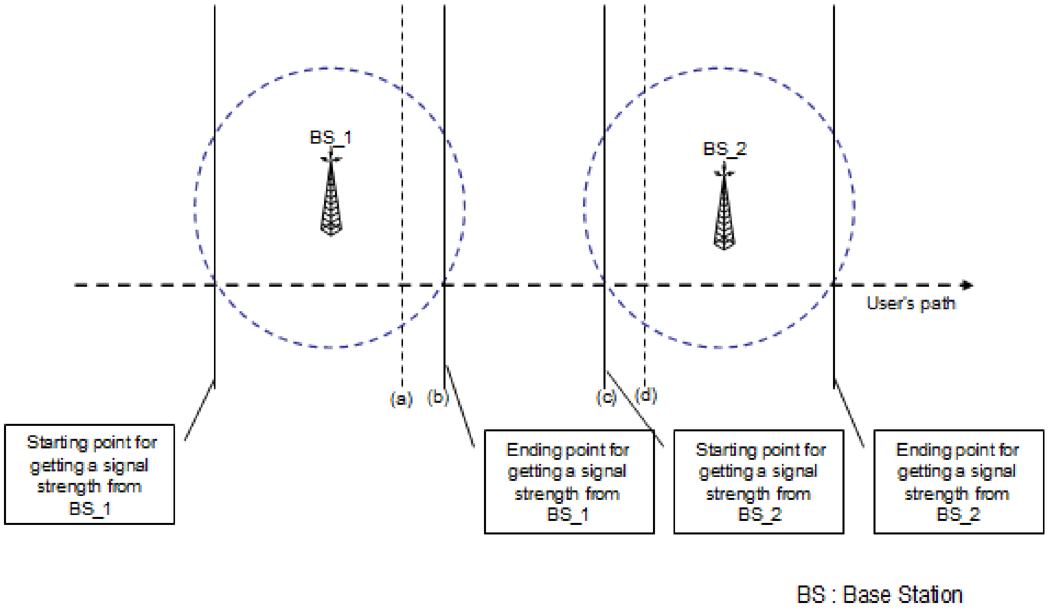

BS_1
BS_2
User's path
(a)
(b)
(c)
(d)
Starting point for getting a signal strength from BS_1
Ending point for getting a signal strength from BS_1
Starting point for getting a signal strength from BS_2
Ending point for getting a signal strength from BS_2
BS : Base Station

Figure 1. A gradual network disruption scenario

Since network disruption could be predicted based on some context, mobile application services could be prepared in case of wireless network disconnections. Therefore, appropriate mechanisms are needed for preparing persistent and/or resumable mobile application services before the current wireless access network is lost.

4. System Overview

We could apply two kinds of strategies, an eager strategy and a lazy strategy, on network system integration to utilize resources effectively. The eager strategy means always preparing and taking actions for possible network disruptions as soon as a mobile device is turned on. The lazy strategy means not preparing and taking actions for possible network disruptions until it is sure that disruption will happen soon. For the eager strategy, we do not need to detect when disruption will happen, but for the lazy strategy, we do need it.

If we adopt the eager strategy for mobile multimedia application services and mobile web application services, there would be network overhead/congestion and undesirable resource consuming on the mobile device. Since network disruptions are not occurred constantly and resources such as mobile device and network bandwidth are limited, it would be better to adopt the lazy strategy. Only the eager strategy could be applied to mobile interactive application services, since they always need to be communicating to the third party server through an intermediary server, which always keeps session information, for network disruption.

In the previous work[10,11], figure 2 illustrates the general model for Mobile Intelligent Server (MIS) [11] -based architecture. MIS is located in the fixed network and has reliable connections to the application server. When the disruption in the wireless access network occurs, the middleware on the mobile device will try to make the mobile application run as long as possible with limited resources until reconnection, and the MIS still collects data from the application server for future update to the mobile device when the network is reconnected between the mobile device and the MIS in case of mobile multimedia application services.

For mobile multimedia application (e.g. video) services, the MIS sets higher sending rate when decision is made based on context information so that the mobile device may reserve more data before network disruption.

For mobile interactive application (e.g. game) services, the MIS makes mobile interactive services be recorded using recording engine during disconnection period and updated in the mobile device after reconnection.

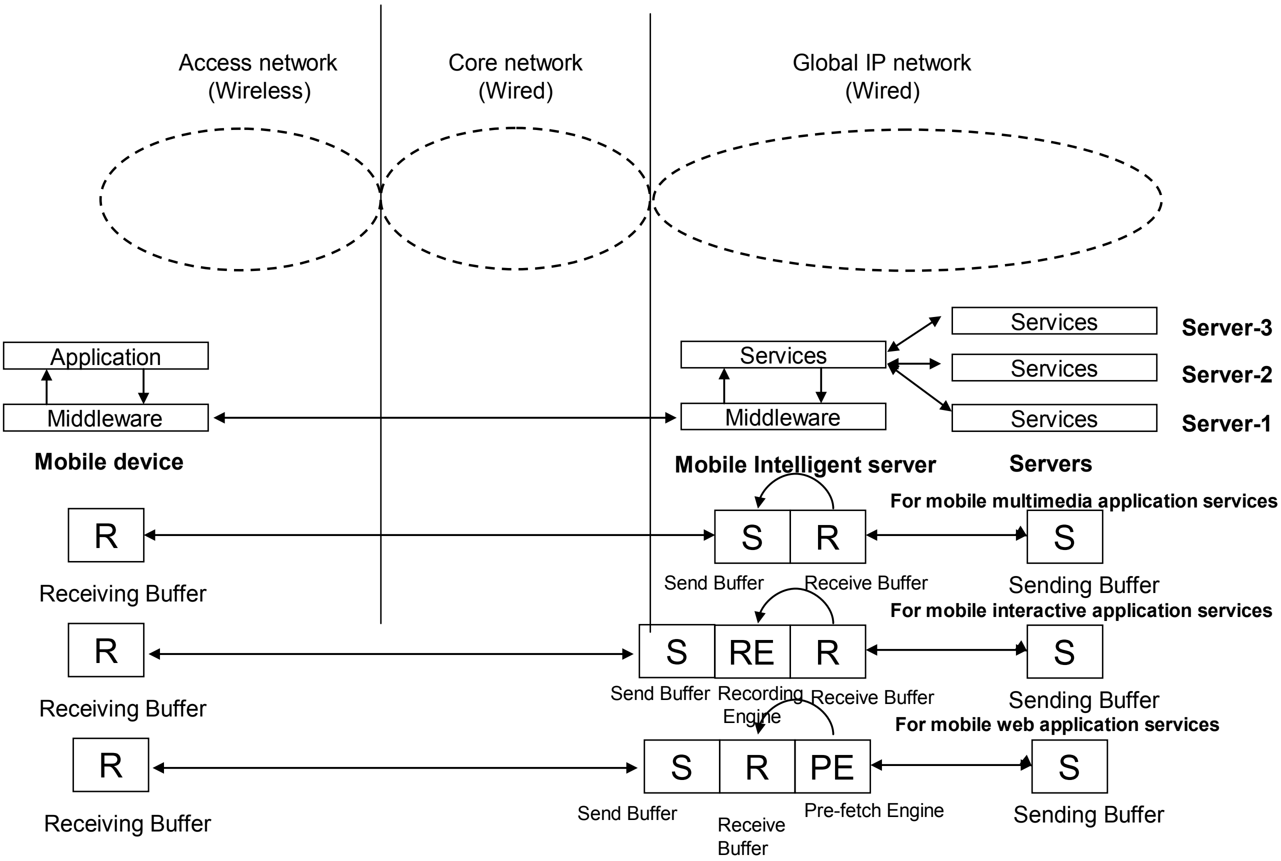


Figure 2. A general model for the middleware solution

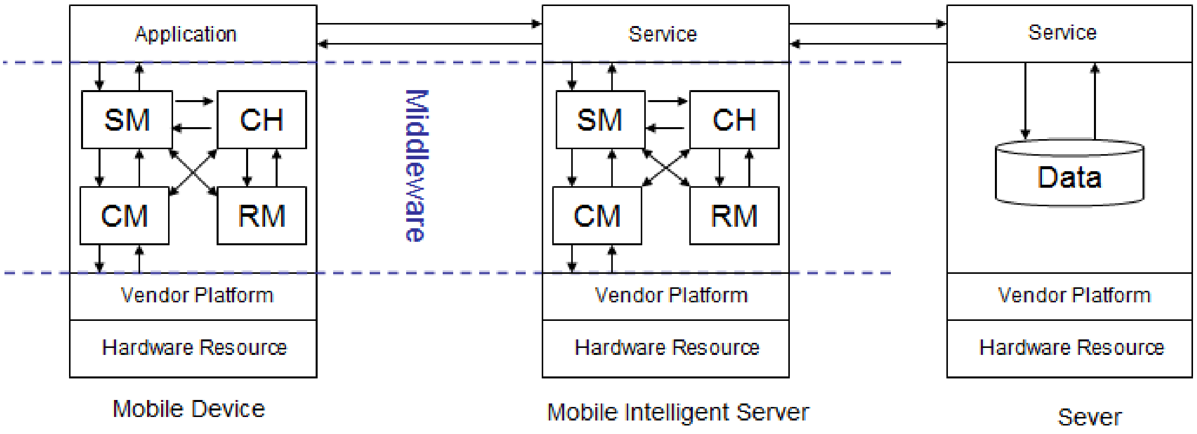


Figure 3. The architecture of the middleware services

For mobile interactive application (e.g. web) services, the MIS makes the first level web pages be pre-patched using pre-patch engine when decision is made based on context information in order to make mobile device be run able during disconnection period. Therefore, there are three main processes for completing our proposed mechanism.

Figure 3 illustrates the architecture of the middleware services. Each of four components in the middleware has a different responsibility. As mentioned in[4], the Service Manager (SM) supports for different mobile application services. The Context Handler (CH) supports decision making based on context information. The Connection Manager (CM) manages network connectivity. The Resource Manager (RM) manages required resources for mobile application services.

5. Context Aware Decisions

As we stated in[11], the proposed context aware decision is for decision making on the question how we are able to know that a device is getting out of coverage area and going to be disconnected for gradual network disconnection. Since we could obtain the dynamic values of signal strength and data rate (device receiving data rate) from the network interface of mobile device as the movement of mobile device and they are inverse proportion with distance from base station or access point, they are possible candidates.

But there are other factors for reducing device receiving data rate[12] such as packet dropping at receiver, interference, hardware quirk, link misalignment, Medium Access Control (MAC) level, and Acknowledgement (ACK) timeout. Therefore, only signal strength could be used for the first required decision.

Therefore, signal strength is used for context aware decision with incremental method in order to avoid momentary fading or shadowing. Since signal strength is usually used for handoff process, our simple decision using signal strength could be compared and referred with the usage of handoff process as shown in figure 4.

While the mobile device is moving from one network range A to another network range B, handoff is usually made at the point, which is an intersecting point between decreasing signal level at range A and increasing signal level at rang B.

For our case, there is no available network while the mobile device is moving form one network range A to another network range B as shown in figure 5. When the signal level A reaches to handoff threshold as the device moves, there has to some proper preparation for network disconnection.

Therefore, in order to make proper preparation for gradual network disconnection, we need two context aware decisions. With regard to the first context aware decision, we have to decide whether a device is getting out of coverage area and is going to be disconnected. We apply incremental method with signal strength. The decision only with signal strength is called simple decision. Table 1 shows context information for simple decision making.

Table 1. Context information for simple decision making

Context	Dimension	Range	Abbreviation
Network	Signal strength	1~10	N_SS

The process of simple decision making is as follows.

If $N\_SS_i - N\_SS_{i-1} > 0$, let $K_i = 1$, where N_SS_i is network signal strength at time i.

else if, $N\_SS_i - N\_SS_{i-1} < 0$, let $K_i = -1$

else $K_i = 0$

Let $inc(N\_SS_{t-n})$ be the sum of K_i for some period time (t-n) from the start time (n) of application to current time (t).

Let $inc(N\_SS_{T-t})$ be the sum of K_i for some period time (T-t) from the current time (t) to the running time (T). In this thesis, the running time (T) is defined as the arbitrarily chosen ongoing time from current time (t) for decision making. However, T has to be chosen by previous individual experience for proper decision making.

When N_SS_t is less than or equal to handoff threshold and $inc(N\_SS_{t-n})$ is less than zero, it indicates that the network signal strength is going to be decreasing if $inc(N\_SS_{T-t})$ becomes less than 0. Therefore, we could predict that network disruption might occur.

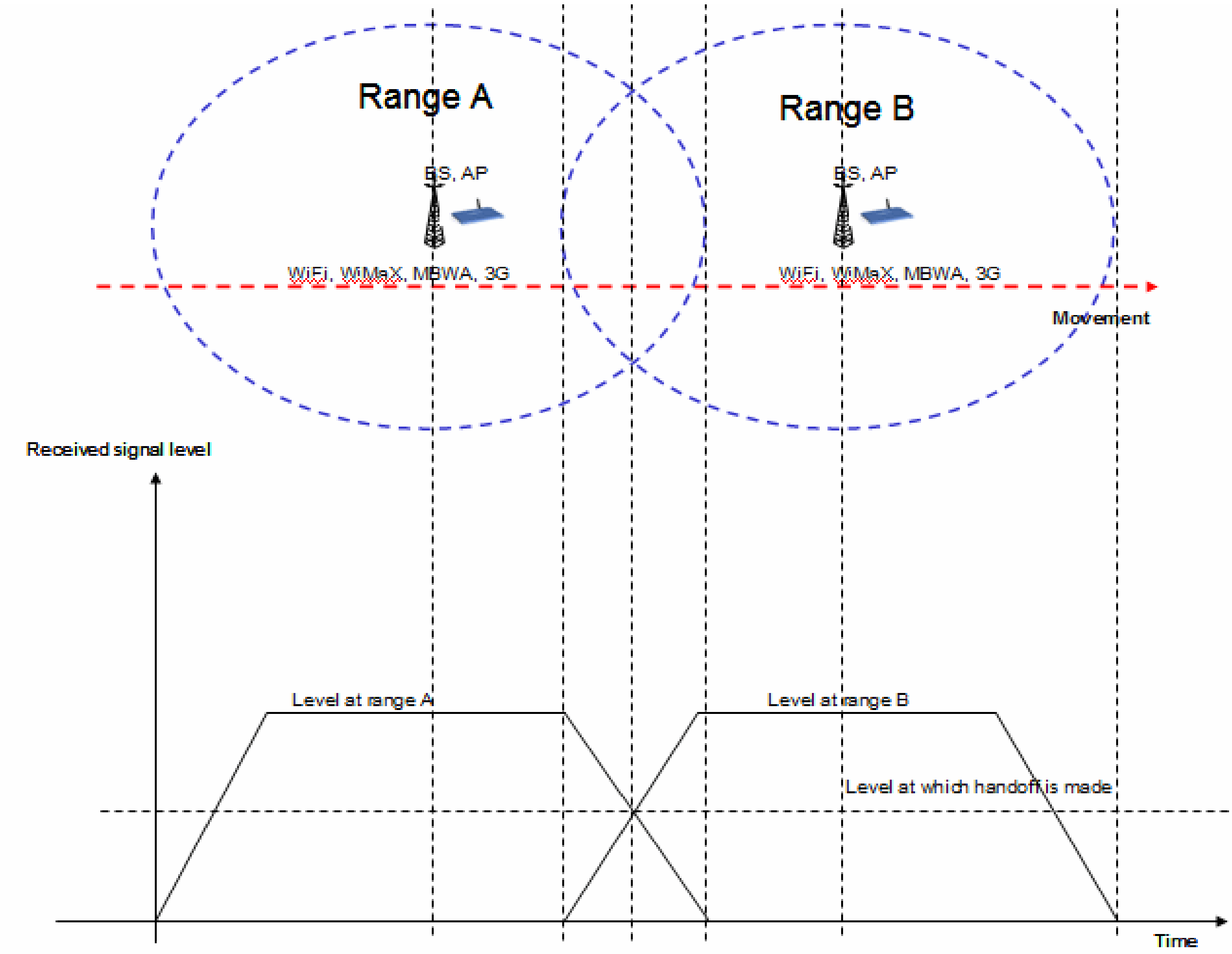


Figure 4. Handoff case using signal strength

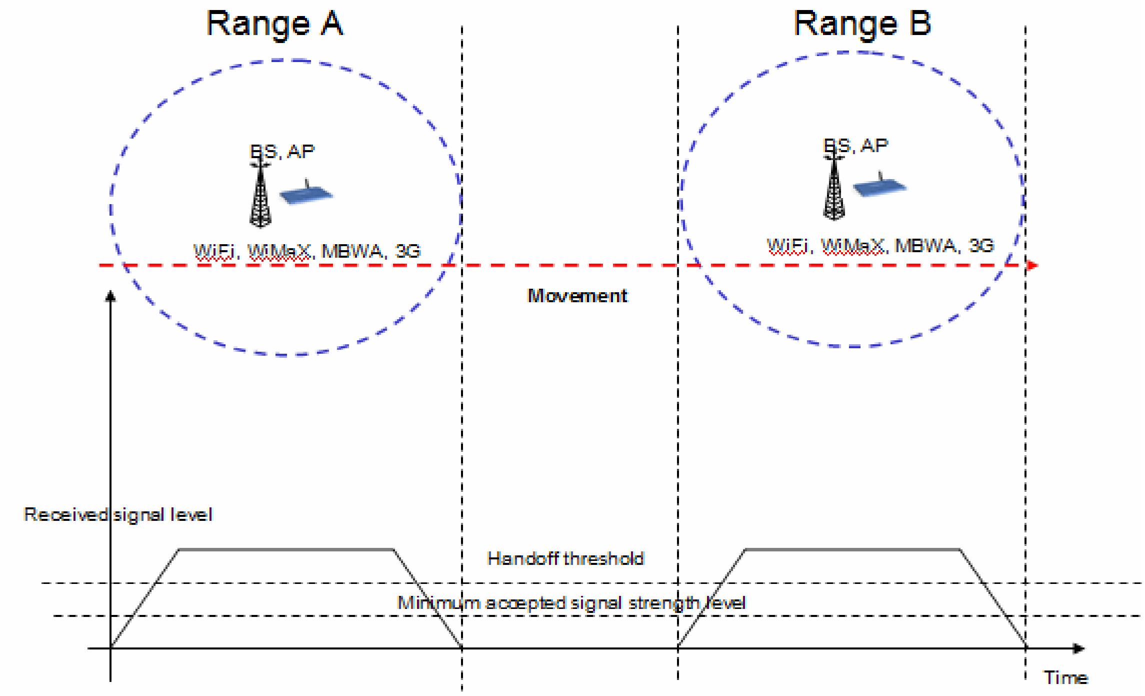


Figure 5. Unavailable handoff case

Therefore, Step 1. Finding $inc(N\_SS_{t\text{-}n})$ and $inc(N\_SS_{T\text{-}t})$

$inc(N\_SS_{t\text{-}n}) = \sum_{i=n}^{t} k_i$, where t= the current time (second) from the start time (n) of application.

$inc(N\_SS_{T\text{-}t}) = \sum_{i=t}^{T} k_i$, where T= the running time (second) from time (t).

Step 2. Making simple decision

Let ND be the network disruption.

When N_SS_t <=handoff threshold and $inc(N\_SS_{t\text{-}n}) < 0$,

ND = $(inc(N\_SS_{T\text{-}t}) < 0)$

Step 3. Preparing network disruption

Requesting more data until N_SS_t <= minimum acceptable signal strength.

Figure 6 provides an example for simple decision making using illustration on a timeline the relative positions of n, t and T. Suppose the handoff threshold is 6. In this example, we have N_SS_t < handoff threshold and $inc(N\_SS_{t\text{-}n}) < 0$ for decision making step 1. Also, we have $inc(N\_SS_{T\text{-}t}) < 0$ for decision making step 2. Therefore, we predict that a gradual network disruption will occur.

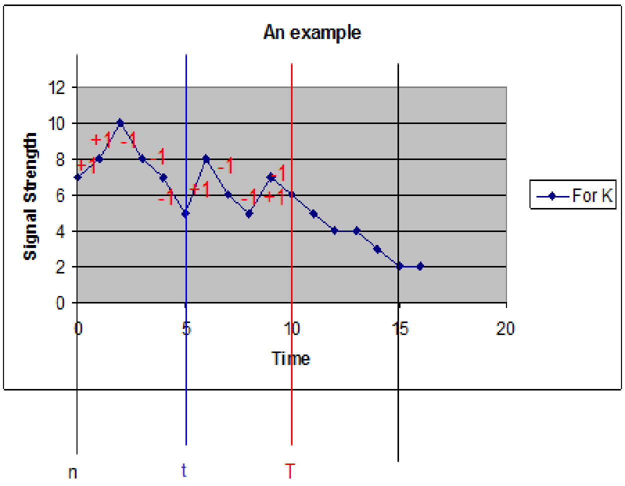


Figure 6. An example for simple decision making

With regard to the second context aware decision on mobile application services, we have to decide when the device should start to make preparation for gradual network disconnection based on network type, application bit rate, device type, device speed, and signal strength. The decision with above context information is called moderate decision. Since the coverage range is different between WiFi network and cellular network, network type has to be considered appropriately. For example, since WiFi network has shorter coverage range than cellular network, the device on movement in WiFi network needs earlier time than in cellular network for preparing network disconnection. Multimedia files are usually encoded at proper bit rate based on estimated/target network bandwidth or throughput for effective playback on mobile device. Therefore, application bit rate has to be considered appropriately. For example, applications with higher application bit rate need to prepare network disconnection earlier. Since each mobile device has different capacity for caching multimedia files, device type has to be considered appropriately. For example, mobile devices with high capacity need to prepare network disconnection earlier. Since the question how early the mobile device is getting out of coverage range is important fact for the second required decision, device speed has to be considered appropriately. For example, faster mobile device need to prepare network disconnection earlier. Table 2 shows context information for moderate decision making.

Table 2. Contexts for moderate decision making

Context	Dimension	Range	Abbreviation
Network	Signal strength	1~10	N_SS
	Type	WiFi=1, 3G=2	N_T
Application	Bit rate	Normal=1, Fast=2	A_BR
Device	Type	Normal=1, High=2	D_T
	Speed	Normal=1, Fast=2	D_S

Since N_T is directly proportional to the running time T, $T' = T^{*}\ N\_T$ should be applied, where T' is adjusted T with the value of contexts for the proper interval of comparison.

Since A_BR, D_T, and D_S are inversely proportional to the running time T, $T' = T^{*}\ (1/A\_BR)^{*}(1/D\_T)^{*}(1/D\_S)$ should be applied for the proper interval of comparison.

Therefore, $T' = T^{*}\ N\_T^{*}(1/A\_BR)^{*}(1/D\_T)^{*}(1/D\_S)$.

6. Experiments

We present detailed experiments, performance results and observations of three mobile application services based on data communication pattern: mobile multimedia application services, mobile web application services and mobile interactive (game) application services. This chapter includes a complete description of experiments for the proposed approach and presents the observed measurements.

As shown in figure 7, we have two test environments for our experiments. The graph (a) and (b) in figure 7 illustrate the outline of experimental environment for obtaining signal strength for verifying decision making algorithms. We obtain various actual data of WiFi signal strength from various areas. We use software named WirelessMon (version 4.0) for the signal strength measurement. The arrows indicate the approximate moving path. Experimental information such as time, channel, percentage (%) of signal strength and signal strength (dBm) is described in appendix C. The graph (b) in figure 7 illustrates the outline of experimental environment for verifying system effectiveness. We use one physical WiFi network with gradual network disruption due to the movements of the user since the purpose of our middleware system is providing mobile application services effectively during disconnection period.

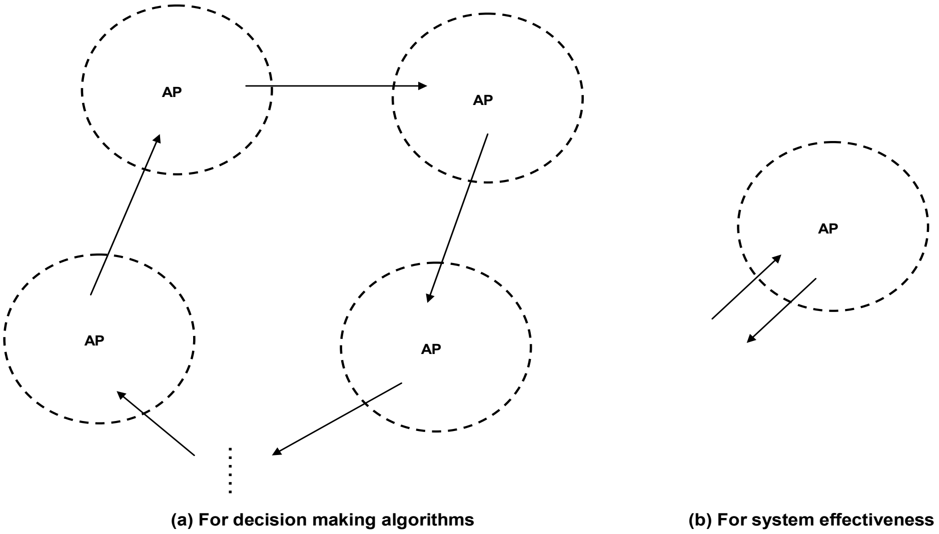
As mentioned before, our design and implementation target to disconnection tolerant mobile application services.

For the actual detailed graph (b) in figure 7, the mobile client uses an HTC Dev Phone and a Samsung Galaxy S that run on Android platform version 1.6 and version 2.1-update respectively. The MIS uses an IBM ThinkPad (T43) that runs on Ubuntu 8.04 (Hardy Heron). Since Android supported multimedia formats (video) are H.263, H.264 AVC and MPEG-4 SP, we decided to use a third party server for mobile multimedia application services on the HP Compaq nx 6320 with Windows XP. The Samsung Galaxy S was used for the experiment on the comparison of mobile multimedia application services with MIS/middleware and without MIS/middleware. For other experiments, the HTC Dev phone was used.

Figure 8 illustrates the overview of the experimental environment.

We designed the prototype of mobile application services to demonstrate the feasibility of applying our proposed techniques. We implemented the mobile services delivery using HTTP/TCP due to their widespread adoption for mobile devices such as mobile handsets and flexibility in implementation.

For mobile multimedia application services, we analyze multimedia data (.mp4/11.3M) received without decision and with context aware decision in order to show how effectively and efficiently we could get data before network disconnection with higher sending rate. Figure 9 illustrates comparison of received data based on sending rate using simple decisions from MIS.



(a) For decision making algorithms

(b) For system effectiveness

Figure 7. Testing environments

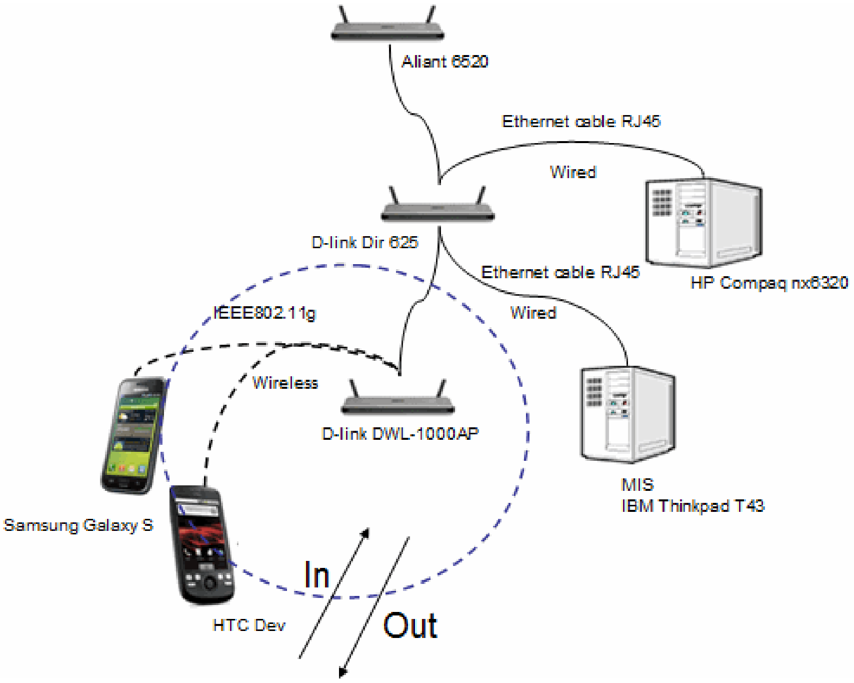


Figure 8. Overview of experimental environment

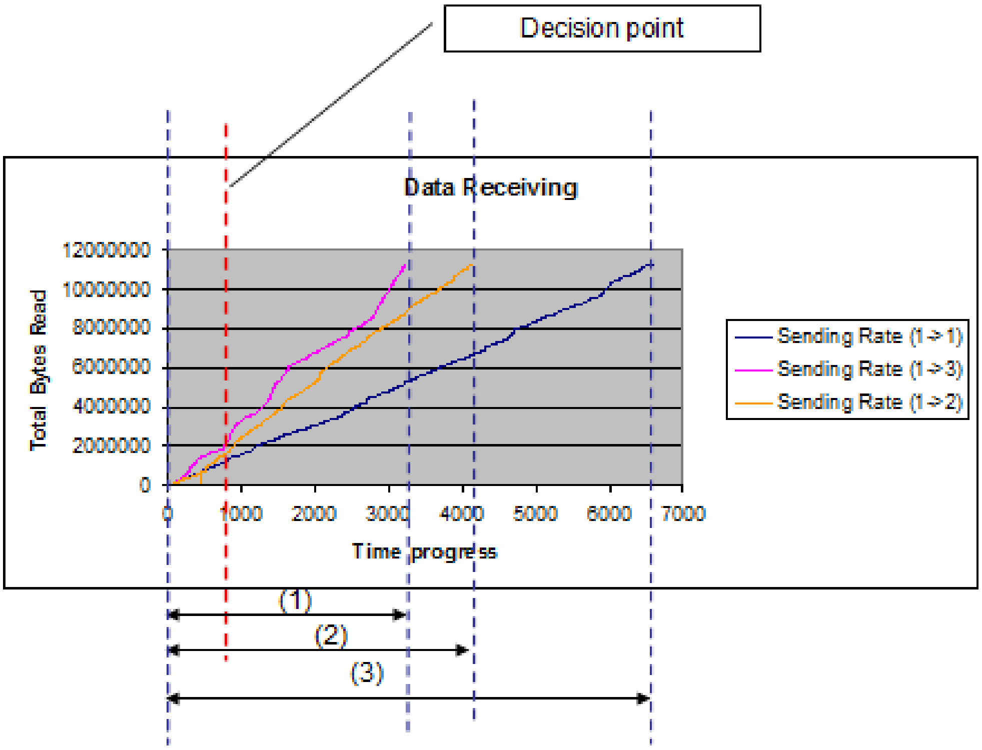


Figure 9. Comparison of received data using simple decisions

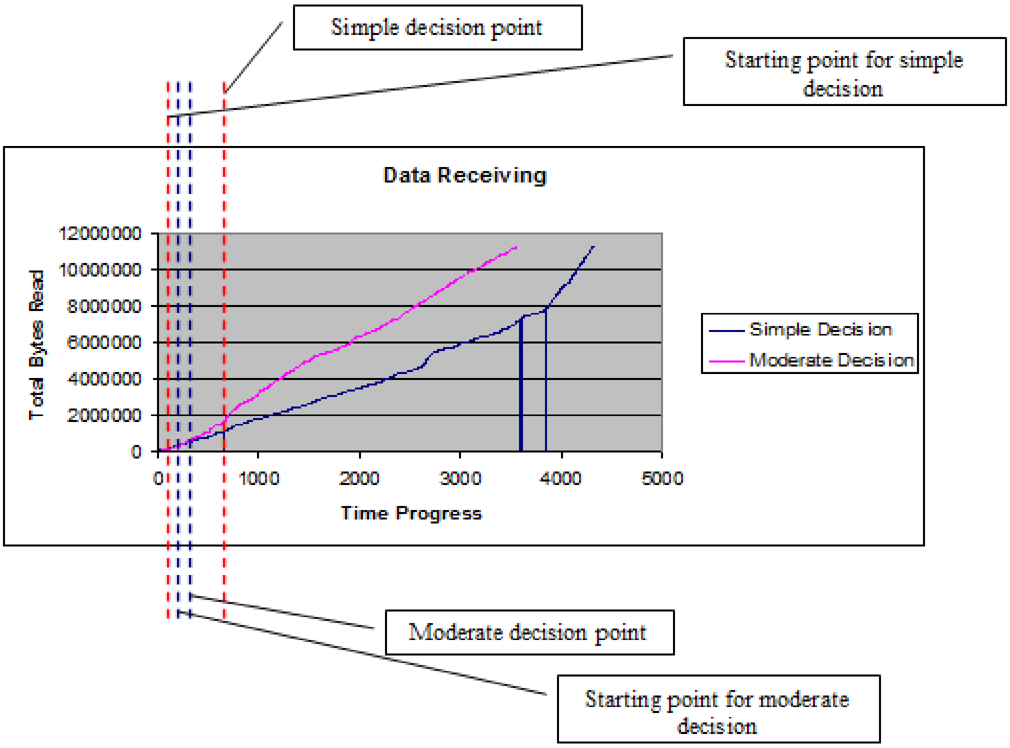


Figure 10. Comparison of received data using moderate decisions

If the sending rate is higher, the total data transfer time is shorter as we expected. Therefore, we could reserve more data before network disconnection if simple decision is made based on sending rate of the MIS. Note that a buffer overflow on mobile device is not considered for our experiments although it can be controlled within our proposed system.

Figure 10 illustrates the comparison of decision making time for simple decision and moderate decision when the device speed has been changed to 2 (fast). The decision making time has been reduced with a fast device speed.

For our experiments, we have compared the below tasks for a short (30 seconds, 60 seconds), a medium (90 seconds), and a long (120 seconds) network disconnection with MIS/middleware and without MIS/middleware.

1. Total bytes received before network disconnection with MIS and without MIS,

2. Total duration of playback during disconnection period with MIS and without MIS,

3. The downloading position of a media file after network reconnection,

4. The resuming time of a media file after network reconnection (ms)

In order to obtain the same exact position where the network is disconnected for appropriate comparison on each experiment case, we set 60 seconds after starting multimedia streaming for the network to be disconnected. Also, we set 30 seconds after starting multimedia streaming for the decision point with MIS. Note that decision making algorithms are not applied for these experiments becase of the reason mentioned above, but decision making algorithms are applied for previous experiments. For these experiments, we use an .mp4/10.8M file and set the sending rate to 1 (15,630 KB/sec) and sending rate to 2 (31,260 KB/sec), the same as with previous experiments. The range of network bandwidth when these experiments were performed is from 24 Mbps to 52 Mbps. Note that the average total playing time of the file is 215,392 ms and application bit rate encoding is 629 kbps. Table 3 describes the experiment result for the short network disconnection (30 Seconds).

The total duration of playback during a disconnection period without MIS/middleware is 54,645ms. There was not a paused position during the disconnection period with MIS/middleware because the buffered data could cover the disconnection period for playback. When the network was disconnected, the mobile application services without MIS/middleware have kept running for 54,645 ms until the received data had been consumed. When the network is reconnected later, we needed to start to request multimedia again from the beginning. The media file with MIS/middleware was downloaded again from the position at 6,401,897 bytes, which is the stop position with the network disconnection, after the network is reconnected. Figure 11 illustrates the comparison of total bytes received with MIS/middleware and without MIS/middleware for the short network disconnection (30 seconds).

The line (1) indicates the decision point for the change of sending rate and the period between lines (2) and (3) indicates the disconnection period. Table 4 describes the experiment result for the short network disconnection (60 seconds).

The total duration of playback during a disconnection period without MIS/middleware is 54,645ms. There was not a paused position during disconnection period with MIS/middleware because the buffered data could cover the disconnection period for playback. When the network was disconnected, the mobile application services without MIS/middleware kept running for 54,645ms until the received data had been consumed. When the network was reconnected later, we needed to start to request multimedia again from the beginning. The media file with MIS/middleware was downloaded again from the position at 6,473,805 bytes, which is the stop position with the network disconnection, after the network is reconnected. Figure 12 illustrates comparison of total bytes received with MIS/middleware and without MIS/middleware for the short network disconnection (60 Seconds).

The line (1) indicates the decision point for the change of sending rate and the period between lines (2) and (3) indicates the disconnection period. Table 5 describes the experiment result for the medium network disconnection (90 seconds).

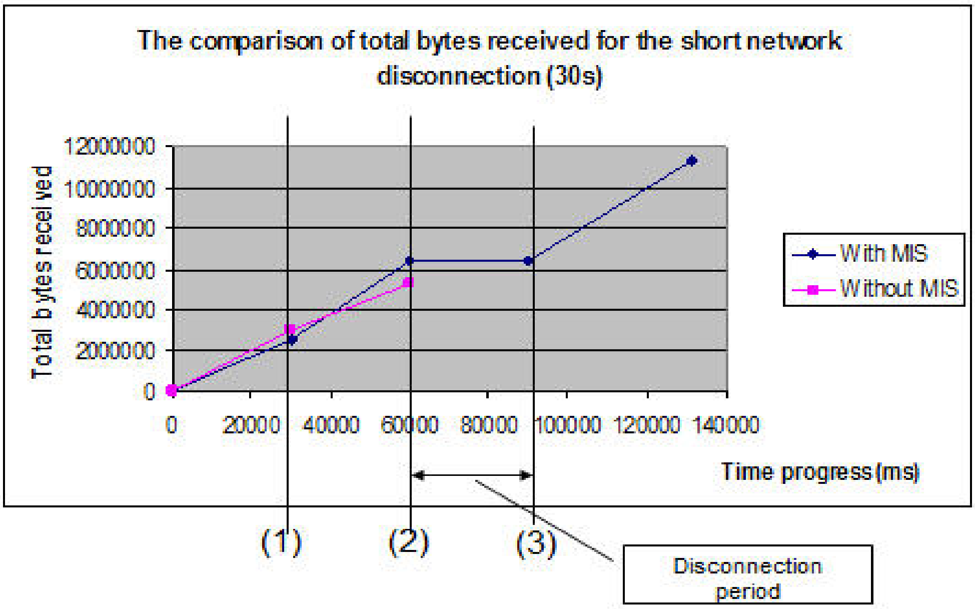


Figure 11. The comparison of total bytes received for the short network disconnection (30s)

Table 3. Experiment result for the short network disconnection (30 S)

Without MIS/middleware				With MIS/middleware			
(1)	(2)	(3)	(4)	(1)	(2)	(3)	(4)
5,316,881	54,645	Beginning	NA	6,401,897	> 30 S	6,401,897	NA

(1): Total bytes received before network disconnection
(2): Total duration of playback during disconnection period (ms)
(3): The downloading position of media file after network reconnection
(4): The resuming time of media file after network reconnection (ms)

Table 4. Experiment result for the long network disconnection (60 S)

Without MIS/middleware				With MIS/middleware			
(1)	(2)	(3)	(4)	(1)	(2)	(3)	(4)
5,316,881	54,645	Beginning	NA	6,473,805	> 60 S	6,473,805	NA

(1): Total bytes received before network disconnection
(2): Total duration of playback during disconnection period (ms)
(3): The downloading position of the media file after network reconnection
(4): The resuming time of the media file after network reconnection (ms)

Table 5. Experiment result for the medium network disconnection (90 S)

Without MIS/middleware				With MIS/middleware			
(1)	(2)	(3)	(4)	(1)	(2)	(3)	(4)
5,316,881	54,645	Beginning	NA	7,281,809	87,378	7,281,809	16,590

(1): Total bytes received before network disconnection
(2): Total duration of playback during disconnection period (ms)
(3): The downloading position of the media file after network reconnection
(4): The resuming time of the media file after network reconnection (ms)

Table 6. Experiment result for the medium network disconnection (120 S)

Without MIS/middleware				With MIS/middleware			
(1)	(2)	(3)	(4)	(1)	(2)	(3)	(4)
5,316,881	54,645	Beginning	NA	6,836,153	77,044	6,836,153	24,128

(1): Total bytes received before network disconnection
(2): Total duration of playback during disconnection period (ms)
(3): The downloading position of the media file after network reconnection
(4): The resuming time of the media file after network reconnection (ms)

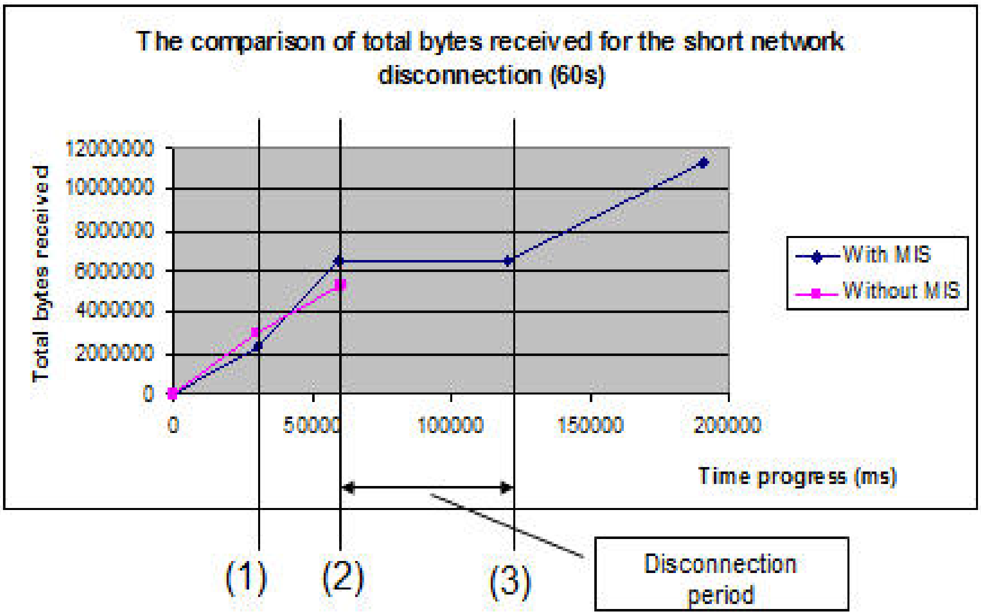


Figure 12. The comparison of total bytes received for the short network disconnection (60s)

The total duration of playback during a disconnection period without MIS/middleware is 54,645 ms whereas the total duration of playback during disconnection period with MIS/middleware is 87,378 ms. When the network was disconnected, the mobile application services without MIS/middleware have kept running for 54,645 ms. When the network is reconnected later, we needed to start to request multimedia again from the beginning without MIS/middleware. The media file with MIS/middleware was downloaded again from the position at 7,281,809 bytes, which is the stop position with the network disconnection, after the network is reconnected. The starting time of the media file after the network reconnected is 16,590 ms. Figure 13 illustrates the comparison of total bytes received with MIS/middleware and without MIS/middleware for the medium network disconnection (90 seconds)

The line (1) indicates the decision point for the change of sending rate and the period between lines (2) and (3) indicates the disconnection period. Table 6 describes the experiment result for the long network disconnection (120 Seconds).

The total duration of playback during the disconnection period without MIS/middleware is 54,645 ms whereas the total duration of playback during disconnection period with MIS/middleware is 77,044 ms. When the network was disconnected, the mobile application services without MIS/middleware kept running for 54,645 ms. When the network reconnected later, we needed to start to request multimedia again from the beginning without MIS/middleware. The media file with MIS/middleware was downloaded again from the position at 6,836,153 bytes, which is the stop position with the network disconnection, after the network is reconnected. The starting time of the media file after the network reconnected is 24,128 ms. Figure 14 illustrates comparison of total bytes received with MIS/middleware and without MIS/middleware for the medium network disconnection (120 seconds).

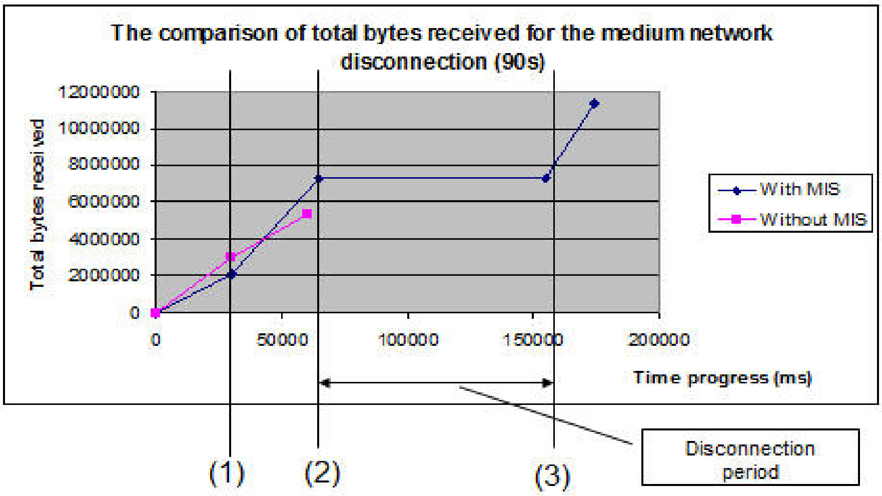


Figure 13. The comparison of total bytes received for the medium network disconnection (90s)

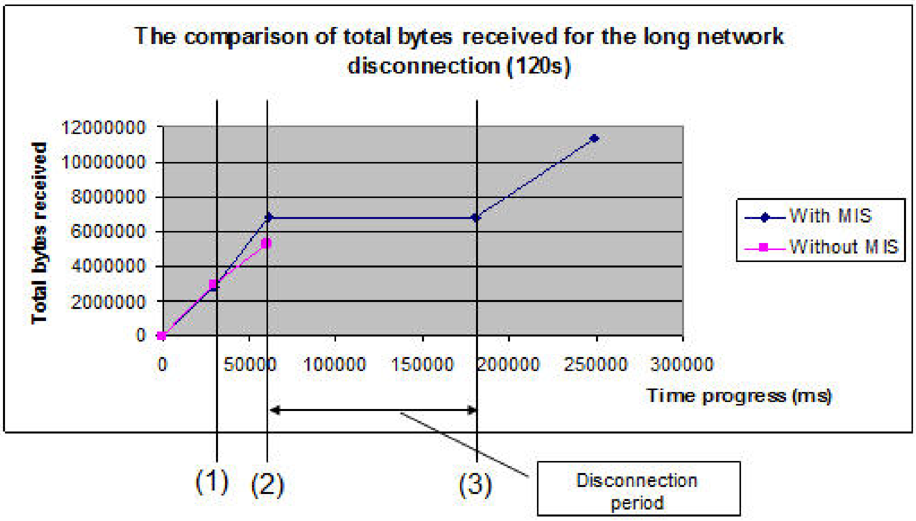
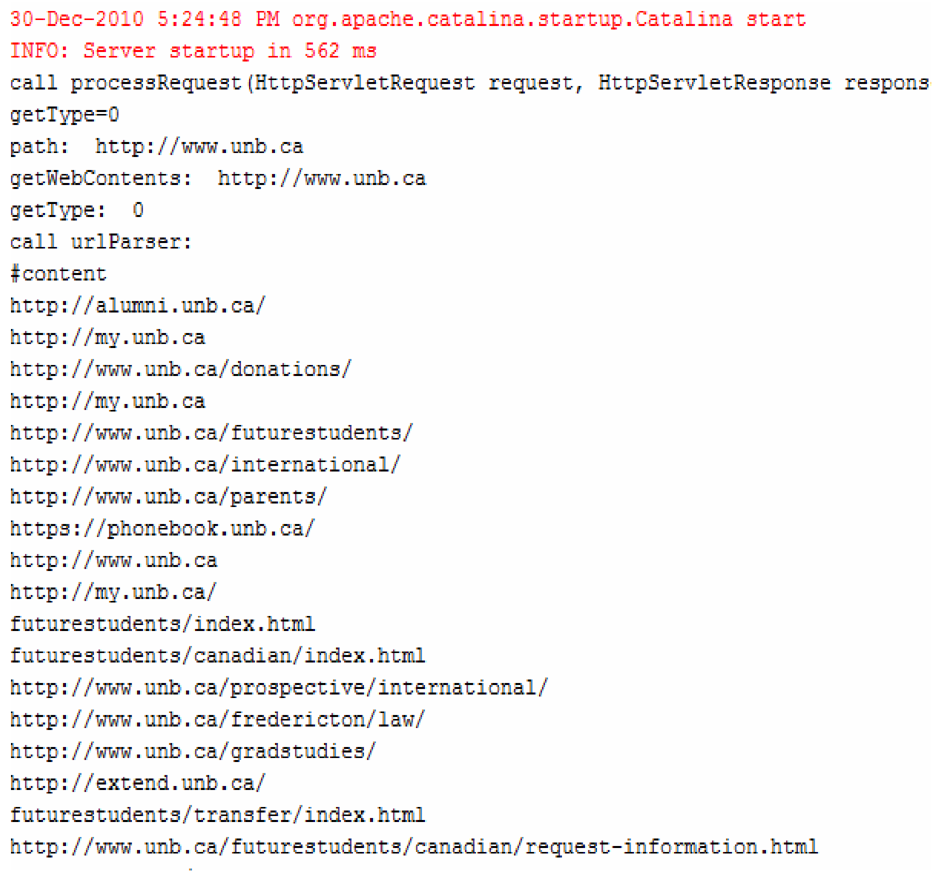
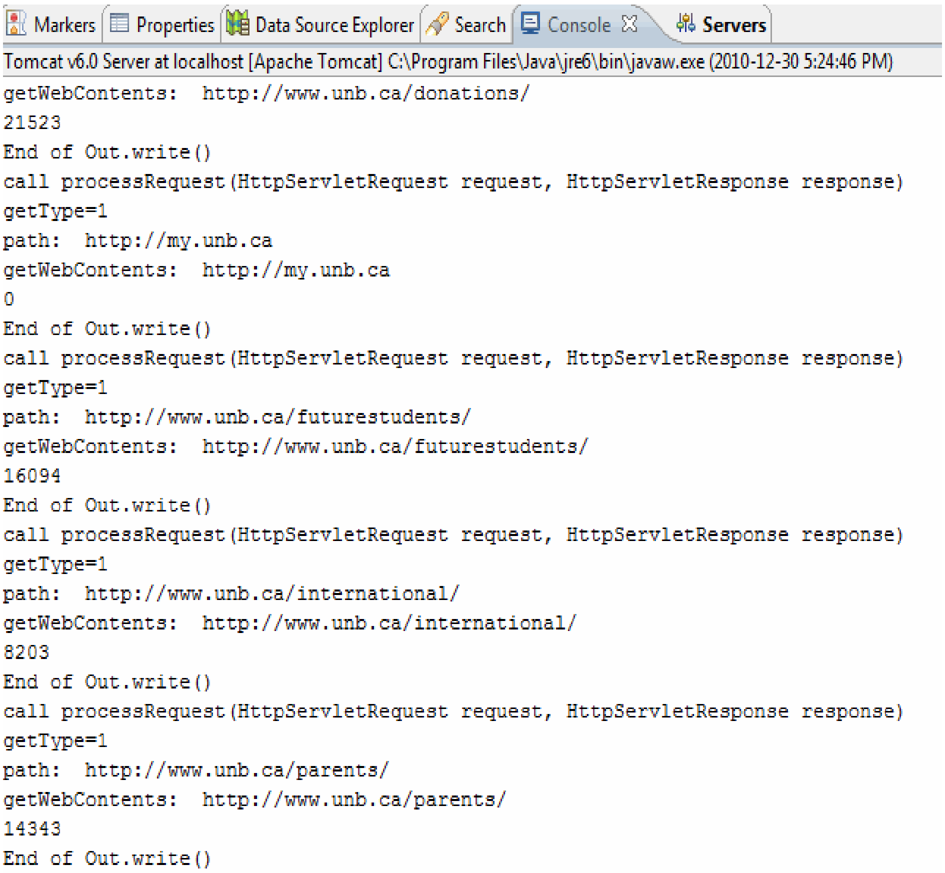


Figure 14. The comparison of total bytes received for the long network disconnection (120s)



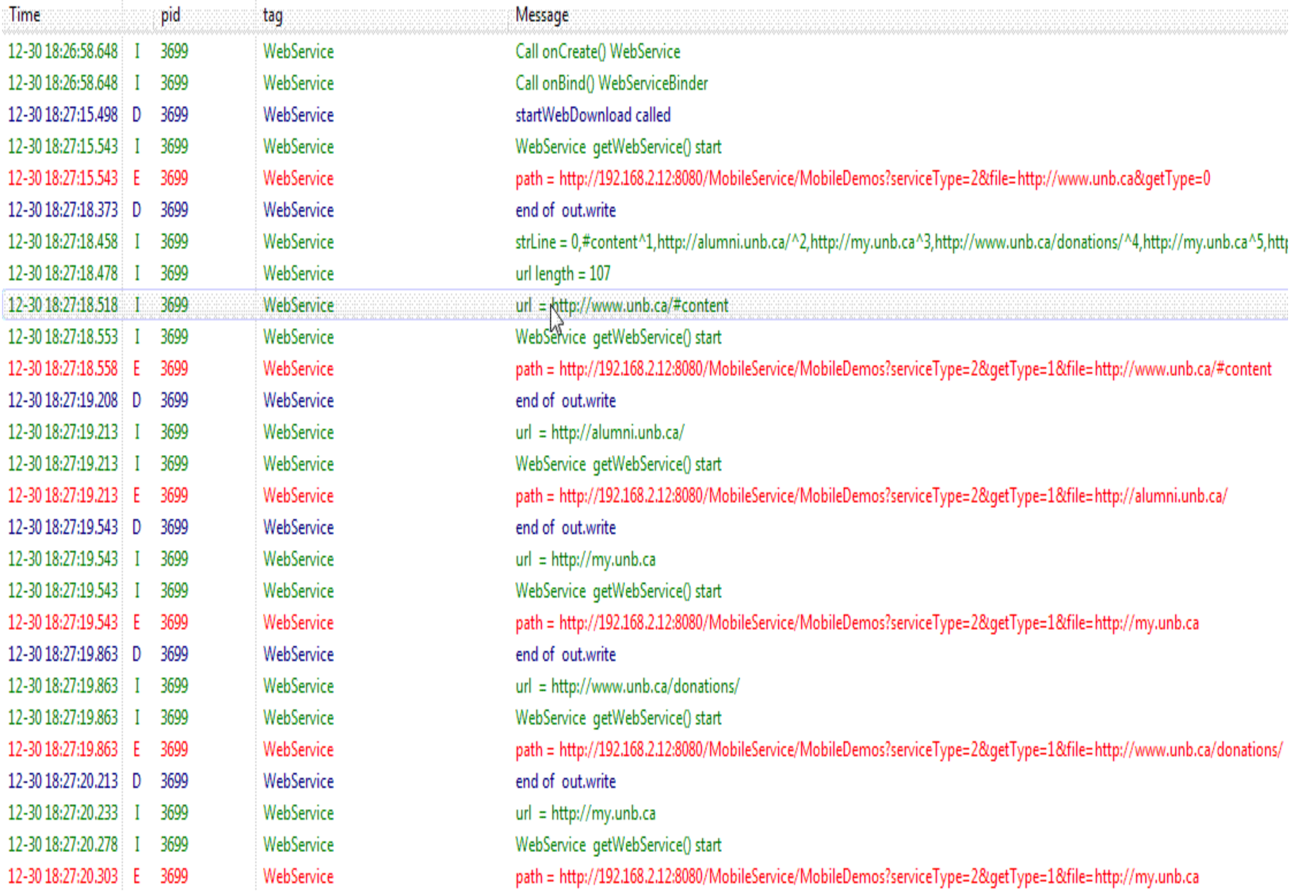
```
30-Dec-2010 5:24:48 PM org.apache.catalina.startup.Catalina start
INFO: Server startup in 562 ms
call processRequest(HttpServletRequest request, HttpServletResponse respons
getType=0
path:  http://www.unb.ca
getWebContents:  http://www.unb.ca
getType:  0
call urlParser:
#content
http://alumni.unb.ca/
http://my.unb.ca
http://www.unb.ca/donations/
http://my.unb.ca
http://www.unb.ca/futurestudents/
http://www.unb.ca/international/
http://www.unb.ca/parents/
https://phonebook.unb.ca/
http://www.unb.ca
http://my.unb.ca/
futurestudents/index.html
futurestudents/canadian/index.html
http://www.unb.ca/prospective/international/
http://www.unb.ca/fredericton/law/
http://www.unb.ca/gradstudies/
http://extend.unb.ca/
futurestudents/transfer/index.html
http://www.unb.ca/futurestudents/canadian/request-information.html
```

Figure 15. The parsing result in MIS



```
Tomcat v6.0 Server at localhost [Apache Tomcat] C:\Program Files\Java\jre6\bin\javaw.exe (2010-12-30 5:24:46 PM)
getWebContents:  http://www.unb.ca/donations/
21523
End of Out.write()
call processRequest(HttpServletRequest request, HttpServletResponse response)
getType=1
path:  http://my.unb.ca
getWebContents:  http://my.unb.ca
0
End of Out.write()
call processRequest(HttpServletRequest request, HttpServletResponse response)
getType=1
path:  http://www.unb.ca/futurestudents/
getWebContents:  http://www.unb.ca/futurestudents/
16094
End of Out.write()
call processRequest(HttpServletRequest request, HttpServletResponse response)
getType=1
path:  http://www.unb.ca/international/
getWebContents:  http://www.unb.ca/international/
8203
End of Out.write()
call processRequest(HttpServletRequest request, HttpServletResponse response)
getType=1
path:  http://www.unb.ca/parents/
getWebContents:  http://www.unb.ca/parents/
14343
End of Out.write()
```

Figure 16. Sending obtained contents to the mobile device



```
Time                 pid    tag          Message
12-30 18:26:58.648 I 3699   WebService   Call onCreate() WebService
12-30 18:26:58.648 I 3699   WebService   Call onBind() WebServiceBinder
12-30 18:27:15.498 D 3699   WebService   startWebDownload called
12-30 18:27:15.543 I 3699   WebService   WebService getWebService() start
12-30 18:27:15.543 E 3699   WebService   path = http://192.168.2.12:8080/MobileService/MobileDemos?serviceType=2&file=http://www.unb.ca&getType=0
12-30 18:27:18.373 D 3699   WebService   end of out.write
12-30 18:27:18.458 I 3699   WebService   strLine = 0,#content^1,http://alumni.unb.ca/^2,http://my.unb.ca^3,http://www.unb.ca/donations/^4,http://my.unb.ca^5,http
12-30 18:27:18.478 I 3699   WebService   url length = 107
12-30 18:27:18.518 I 3699   WebService   url = http://www.unb.ca/#content
12-30 18:27:18.553 I 3699   WebService   WebService getWebService() start
12-30 18:27:18.558 E 3699   WebService   path = http://192.168.2.12:8080/MobileService/MobileDemos?serviceType=2&getType=1&file=http://www.unb.ca/#content
12-30 18:27:19.208 D 3699   WebService   end of out.write
12-30 18:27:19.213 I 3699   WebService   url = http://alumni.unb.ca/
12-30 18:27:19.213 I 3699   WebService   WebService getWebService() start
12-30 18:27:19.213 E 3699   WebService   path = http://192.168.2.12:8080/MobileService/MobileDemos?serviceType=2&getType=1&file=http://alumni.unb.ca/
12-30 18:27:19.543 D 3699   WebService   end of out.write
12-30 18:27:19.543 I 3699   WebService   url = http://my.unb.ca
12-30 18:27:19.543 I 3699   WebService   WebService getWebService() start
12-30 18:27:19.543 E 3699   WebService   path = http://192.168.2.12:8080/MobileService/MobileDemos?serviceType=2&getType=1&file=http://my.unb.ca
12-30 18:27:19.863 D 3699   WebService   end of out.write
12-30 18:27:19.863 I 3699   WebService   url = http://www.unb.ca/donations/
12-30 18:27:19.863 I 3699   WebService   WebService getWebService() start
12-30 18:27:19.863 E 3699   WebService   path = http://192.168.2.12:8080/MobileService/MobileDemos?serviceType=2&getType=1&file=http://www.unb.ca/donations/
12-30 18:27:20.213 D 3699   WebService   end of out.write
12-30 18:27:20.233 I 3699   WebService   url = http://my.unb.ca
12-30 18:27:20.278 I 3699   WebService   WebService getWebService() start
12-30 18:27:20.303 E 3699   WebService   path = http://192.168.2.12:8080/MobileService/MobileDemos?serviceType=2&getType=1&file=http://my.unb.ca
```

Figure 17. Receiving contents in the mobile device

The line (1) indicates the decision point for the change of sending rate and the period between lines (2) and (3) indicates the disconnection period.

For mobile web application services, we use the UNB web site. Figure 15 illustrates URL parsing results for the first sub level of the current web document in the MIS in order to reduce network overhead when the mobile device makes simple decision. It is a sample of the full output.

Figure 16 illustrates the process of sending obtained contents for the parsed URLs in the MIS from the server to the mobile device. It is a sample of the full output.

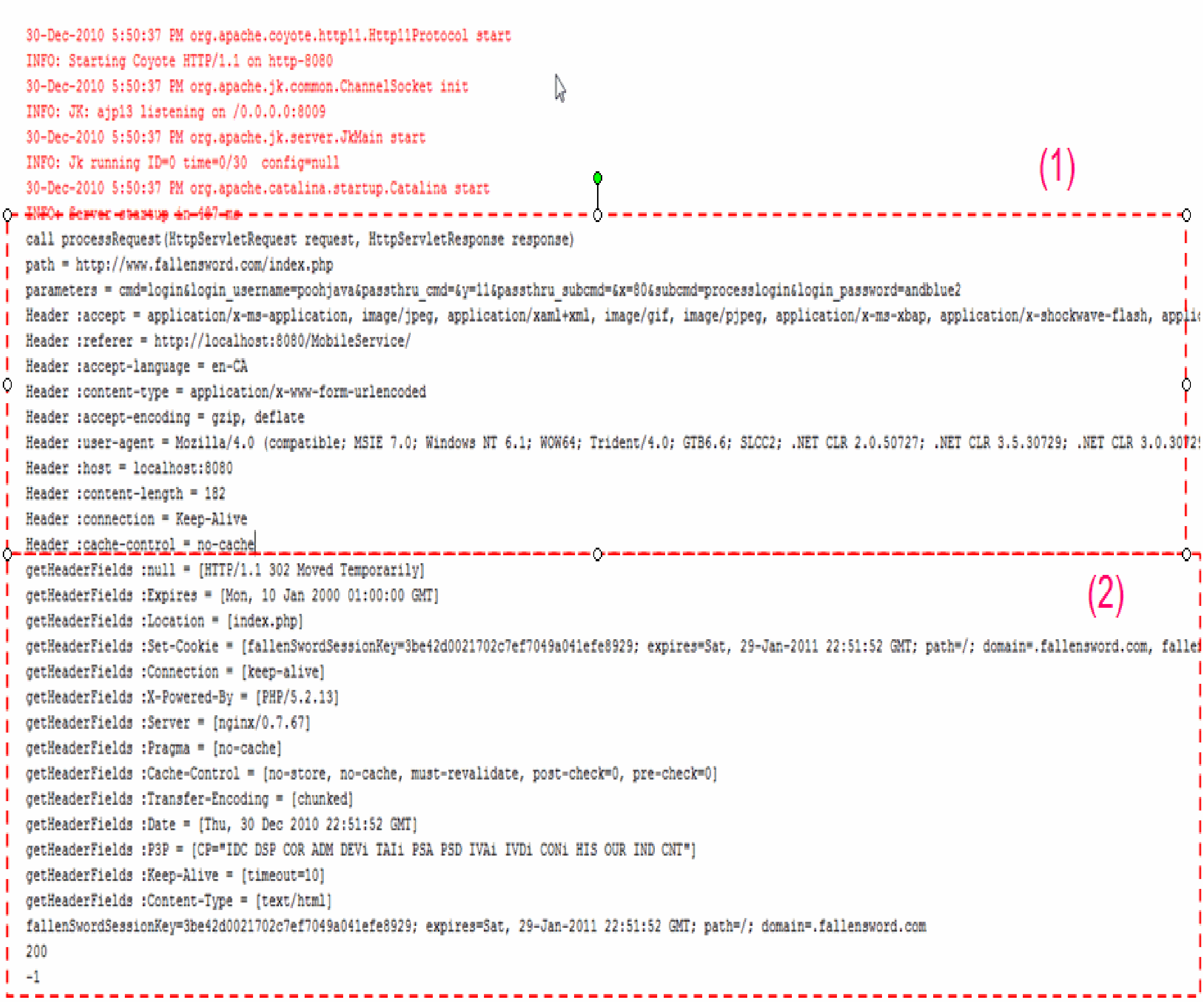
Figure 17 illustrates the process of receiving contents from the MIS to the mobile device. It is a sample of the full output.

As shown in the experiment, the mobile device can obtain the first level of the current web document before network disconnection so that the user may enjoy web services without undesirable termination during a network disconnection period. For some web sites such as www.m.naver.com, we could not obtain the content information for the current web page. There could be many different reasons within the design of these web sites such as redirection.

For interactive (e.g. game) application services, we use public game site http://www.fallensword.com. Figure 18 illustrates the way MIS obtains the information of a current session between a mobile device and server.

Process (1) indicates that the MIS requests interactive (game) service with user name, password and header information provided by the mobile device. Process (2) indicates that the MIS obtains header information including session data for game contents.

Therefore, if there is a simple decision for handling network disruption, the MIS keeps connection with the server based on the session information. After reconnection, the mobile device may resume the interactive service through the MIS.



```
30-Dec-2010 5:50:37 PM org.apache.coyote.http11.Http11Protocol start
INFO: Starting Coyote HTTP/1.1 on http-8080
30-Dec-2010 5:50:37 PM org.apache.jk.common.ChannelSocket init
INFO: JK: ajp13 listening on /0.0.0.0:8009
30-Dec-2010 5:50:37 PM org.apache.jk.server.JkMain start
INFO: Jk running ID=0 time=0/30  config=null
30-Dec-2010 5:50:37 PM org.apache.catalina.startup.Catalina start
INFO: Server startup in 407 ms
call processRequest(HttpServletRequest request, HttpServletResponse response)
path = http://www.fallensword.com/index.php
parameters = cmd=login&login_username=poohjava&passthru_cmd=&y=11&passthru_subcmd=&x=80&subcmd=processlogin&login_password=andblue2
Header :accept = application/x-ms-application, image/jpeg, application/xaml+xml, image/gif, image/pjpeg, application/x-ms-xbap, application/x-shockwave-flash, appli
Header :referer = http://localhost:8080/MobileService/
Header :accept-language = en-CA
Header :content-type = application/x-www-form-urlencoded
Header :accept-encoding = gzip, deflate
Header :user-agent = Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; WOW64; Trident/4.0; GTB6.6; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.3072
Header :host = localhost:8080
Header :content-length = 182
Header :connection = Keep-Alive
Header :cache-control = no-cache
getHeaderFields :null = [HTTP/1.1 302 Moved Temporarily]
getHeaderFields :Expires = [Mon, 10 Jan 2000 01:00:00 GMT]
getHeaderFields :Location = [index.php]
getHeaderFields :Set-Cookie = [fallenSwordSessionKey=3be42d0021702c7ef7049a041efe8929; expires=Sat, 29-Jan-2011 22:51:52 GMT; path=/; domain=.fallensword.com, falle
getHeaderFields :Connection = [keep-alive]
getHeaderFields :X-Powered-By = [PHP/5.2.13]
getHeaderFields :Server = [nginx/0.7.67]
getHeaderFields :Pragma = [no-cache]
getHeaderFields :Cache-Control = [no-store, no-cache, must-revalidate, post-check=0, pre-check=0]
getHeaderFields :Transfer-Encoding = [chunked]
getHeaderFields :Date = [Thu, 30 Dec 2010 22:51:52 GMT]
getHeaderFields :P3P = [CP="IDC DSP COR ADM DEVi TAIi PSA PSD IVAi IVDi CONi HIS OUR IND CNT"]
getHeaderFields :Keep-Alive = [timeout=10]
getHeaderFields :Content-Type = [text/html]
fallenSwordSessionKey=3be42d0021702c7ef7049a041efe8929; expires=Sat, 29-Jan-2011 22:51:52 GMT; path=/; domain=.fallensword.com
200
-1
```

Figure 18. Obtaining the information of the current session

7. Evaluation

We have shown the effectiveness of the proposed middleware services in the previous section since our system is uniquely specialized for mobile application services in case of network disruption.

To verify the accuracy of our proposed (simple, moderate) decision making algorithms for the second objective, we have tested the system with various actual data of WiFi signal strength while a user is moving in and out of the coverage range of access point from various areas. We have a total of 17 testing data sets. We apply the proposed decision making algorithms to each data set. For our research purpose, we consider that the simple or moderate decision is correct when the current signal strength reaches zero after decision is made. We consider that the simple or moderate decision is incorrect when the current signal strength reaches a handoff threshold again after decision made.

The accuracy rate is calculated by (the number of data sets with correct decision / total number of testing data sets) * 100. Since intelligent decision making uses distance between a mobile device and a location that experiences sudden network disruption for the decision making process, the distance is decreased when a user is moving toward the network disruption location. Therefore, it is obvious that the accuracy of intelligent decision making algorithm will be similar to simple and moderate decisions. The simulator developed in Java has been used for this verification.

The process of simulator is as follows:

Step 1: Read each line of signal strength data from a file.

Step 2: Test the proposed algorithm with the data set described in section 4.5 and section 4.6.

Step 3: Make decision

Step 4: Analyze decision accuracy

If there is not enough data due to high speed movement in each data set to test simple decision making, we count that it is an unavailable record. Table 7 indicates the accuracy of simple decision with unavailable data. Table 8 indicates the accuracy of simple decision without unavailable data.

Table 7. The accuracy of simple decision with unavailable data

Correct Data Sets	Incorrect Data Sets	Unavailable Data Sets	Total Data Sets
10	1	6	17
58.8 %	5.8 %	35.2%	100 %

Table 8. The accuracy of simple decision without unavailable data

Correct Data Sets	Incorrect Data Sets	Total Data Sets
10	1	11
90.9 %	9.0%	100 %

We set the device speed to 2 to know the running time *T'* is affected by context information for moderate decision. All records for simple decision are affected by context information. Also, Table 9 indicates the changes of Table 8-1 with moderate decision. Three unavailable data from Table 7 have become available, which means that *T'* is affected by context information, that is, device speed.

Table 9. The change of table 7 with moderate decision

Correct Data Sets	Incorrect Data Sets	Unavailable Data Sets	Total Data Sets
13	1	3	17
76.4 %	5.8 %	17.6%	100 %

Note that the number of data sets used in the evaluation was rather small, e.g. there was only one "incorrect" data set, which may affect the accuracy of the evaluation. In future work, we will try to have more sufficient data sets to verify the accuracy of our proposed (simple, moderate) decision making algorithms appropriately.

8. Concluding Remarks

In this paper, we explore middleware service for disconnection tolerant mobile application services. The proposed approach could play an important role of supporting the service mobility for mobile application services.

For future work, we will further investigate on optimal algorithms to improve accuracy rate for decision making cases and user event patterns for the continuation of mobile interactive application services

ACKNOWLEDGEMENTS

This CNSR research was made possible thanks to support from Bell Canada through its Bell University Laboratories R&D program, and from ACOA through an AIF grant.

REFERENCES

[1] Q. Yuan, I. Cardei and J. Wu, “Predict and Relay: An efficient routing in disruption-tolerant networks” in Proceedings of the 10th ACM international symposium on mobile ad hoc networking and computing, pp. 95-104, 2009.

[2] J. Leguay, T. Friedman and V. Conan, “DTN routing in a mobility pattern space” in Proceedings of the 2005 ACM SIGCOMM workshop on delay-tolerant networking, pp. 276-283, 2005.

[3] J. Burgess, B. Gallagher, D. Jensen, B. N. Levin, “MaxProp: Routing for vehicle-based disruption-tolerant networks” in Proceedings of the 25th IEEE international Conference on Computer Communications , pp. 1-11, 2006.

[4] J. Ott, “Application protocol design considerations for a mobile internet” in Proceedings of the 1st ACM/IEEE international workshop on mobility in the evolving internet architecture, pp. 75-80, 2006.

[5] S. Farrell, V. Cahill, D. Geraphty, I. Humphreys, and P. McDonald, “When TCP breaks: delay-and disruption-tolerant networking” Internet computing, IEEE, Vol. 10, Issue 4, pp. 72-78, Sep. 2006.

[6] D. Salber, A.K. Dey and G.D. Abowd, “The context toolkit: aiding the development of context-enabled applications,” in Proceedings of the SIGCHI Conference on Human Factors *in Computing Systems*, pp. 434-441,1999.

[7] A. Moghadam, S. Srinivasan and H. Schulzrinne, “7DS- A modular platform to develop mobile disruption-tolerant applications,” in Proceedings of the 2nd international conference on next generation mobile applications, services and technologies, pp. 177-183,2008.

[8] A. Moghadam, 7 DS System. [Online], Available: http://www1.cs.columbia.edu/~arezu/7DS/index.html

[9] R. Cheung, G. Yao, J. Cao and A. Chan, “ A fuzzy service adaptation engine for context-aware mobile computing middleware,” International Journal of Pervasive Computing and Communications, Vol. 4, No. 2, pp. 147-165, 2008.

[10] S. Cha, W. Du and B.J. Kurz, “Middleware framework for disconnection tolerant mobile application services,” in 8th annual communication networks and services research conference (CNSR-IEEE), pp. 334-340, 2010.

[11] S. Cha, W. Du, “Context Aware Middleware Services for Disconnection Tolerant Mobile Applications,” in 9th Annual Communication Networks and Research Conference (CNSR-IEEE), pp.145-152,2011.

[12] D. Golchha, “Wi-Fi NetMon : Performance observation, Anomaly detection and diagnosis in long distance Wi-Fi networks” in M.S. thesis, Indian institute of technology Bombay, 2007.